REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	Sept 2001	Final Report, ~~March 2000 – March 2001~~ 01 May 96 - 30 Apr 01

4. TITLE AND SUBTITLE
A Software Architecture For Scalable Simulations on Heterogeneous Networks

5. FUNDING NUMBERS
~~DAAD19-99-1-0017~~
DAAH04-96-1-0083

6. AUTHOR(S)
V. Rego and V. Sunderam

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Purdue University, West Lafayette, IN 47906

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
34798.20-MA-SDI

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT	12 b. DISTRIBUTION CODE
Approved for public release; distribution unlimited.	

13. ABSTRACT (Maximum 200 words)

Our initial research since the inception of this project has established the viability and effectiveness of each of the novel aspects of our approach to massively concurrent computing viz. specialized communications substrate, threads based programming, semi-automatic domain specific parallelization, and failure resilient computing. Experiences with a few, but nonetheless representative, classes of applications has demonstrated the efficacy of our prototype systems, and have established the justifications for building robust production quality versions thereof. In the process, our work has also highlighted several interesting research issues from the computer systems and computer science points of view. We have focused on four layers of a software architecture : domain layers (ParaSol and EcliPSe) and support layers (Ariadne -- for threads, and Conch/Clam for communication). The architecture uses a two-level control hierarchy, exploiting locality and new (UDP-based) network protocols to reduce communication times, enhance functionality and improve performance. The modification involves designating specific processes within the host pool as sub-domain servers(SDS). The first process to be initiated on each subnet automatically instantiates a thread that serves as the primary SDS for that subset of the host pool, with subsequent processes assuming secondary responsibility in the case of failures. Each SDS is responsible for process and thread creation/destruction on its machines, as well as for collecting load and resource availability statistics. By exchanging control messages among themselves, the SDS threads provide hints to the higher layers (EcliPSe, Ariadne} and ParaSol) to enable workload allocation based on balanced distribution of computation and on minimized communication across subnet boundaries.

14. SUBJECT TERMS	15. NUMBER OF PAGES
threads, protocols, subdomain, subnet, udp	3
	16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev.2-89)
Prescribed by ANSI Std. 239-18
298-102

20011113 144

MASTER COPY: PLEASE KEEP THIS "MEMORANDUM OF TRANSMITTAL" BLANK FOR REPRODUCTION PURPOSES. WHEN REPORTS ARE GENERATED UNDER THE ARO SPONSORSHIP, FORWARD A COMPLETED COPY OF THIS FORM WITH EACH REPORT SHIPMENT TO THE ARO. THIS WILL ASSURE PROPER IDENTIFICATION. NOT TO BE USED FOR INTERIM PROGRESS REPORTS; SEE PAGE 2 FOR INTERIM PROGRESS REPORT INSTRUCTIONS.

MEMORANDUM OF TRANSMITTAL

U.S. Army Research Office
ATTN: AMSRL-RO-BI (TR)
P.O. Box 12211
Research Triangle Park, NC 27709-2211

☐ Reprint (Orig + 2 copies)

☐ Technical Report (Orig + 2 copies)

☐ Manuscript (1 copy)

☒ Final Progress Report (Orig + 2 copies)

☐ Related Materials, Abstracts, Theses (1 copy)

CONTRACT/GRANT NUMBER: DAAH04-96-1-0083

REPORT TITLE: A Software Architecture for Scalable Simulations on Heterogeneous Networks

is forwarded for your information.

SUBMITTED FOR PUBLICATION TO (applicable only if report is manuscript):

Sincerely,

Vernon Rego

Over the course of this project, we have built a functional and scalablesoftware architecture distributed computing and simulation, based on a layered domain, user-space threads and lightweight protrocols. We have completed the ParaSol kernel and tested its functionality by coding distinct domain-layers upon this kernel. We are currently working with domain experts to make the interactions between the domain level and the kernel work successfully. To accomplish this, we will re-organize and modularize the computational framework within our software suite, followed by the construction of application domain specific classes. Second, after we found that our communications substrate (Conch) -- which is based on TCP/IP, and does software-based topological routing -- did not provide the adequate performance and functionality to deliver requisite perrformance for communications-bound applications, we designed and built the CLAM communication system, based on the user-space threads in our Ariadne threads library. Our idea was to use active-messages (threads) and UDP/IP multicast to (1). overlap communication and computation, and (2). simultaneously provide several levels of communications service -- including an MPI interface -- so that applications requiring image and audio-visual data may also exploit this computing environment.

We continue to evolve our software infrastructures into robust prototypes with significantly enhanced usability and performance. We have:

1. reimplemented a robust, modular, and extensible threads-based computational framework, and constructed an extensible suite of distinct application-domain libraries (with help from our collaborators), upon the ParaSol kernel,

2. evolved an alternative communications substrate supporting the MPI interface, based on IP multicasting and stateless datagram protocols, to enhance scalability, improve communication performance and reduce setup and other overheads,

3. built an organized system of "sub-domain servers", to enable effective workload partitioning, failure resilience, and critical subdomain-based multicast routing -- leading to improved performance, and

4. provided our collaborators with methodological guidelines for system use, documentation, support for porting and testing applications coded with the help of domain-libraries, and support for demonstrating functionality and performance.

Status

The primary goal of the ACES project was to investigate software architectures for massively concurrent computing in heterogeneous networked environments. In the few years since inception, we have developed several innovative approaches to effective, failure resilient, high performance computing in geographically and administratively distributed environments. Focusing on specific sub-goals, we have built (1) a run-time system (Ariadne) supporting portable, migratable threads, so that we could exploit multithreading in the design of domain-specific libraries; (2) a communications substrate (Conch, and later CLAM) with a process-relocation (for fault-tolerance) capability, and support for logical interconnection topologies in heterogeneous networked environments; (3) a new EcliPSe replications library, this time utilizing support from Conch -- by providing a check-pointing primitive, we have provided semi-transparent fault-tolerance for EcliPSe applications; and (4) built the ParaSol kernel -- to support threads-based computation and simulation (i.e., computations involving

discrete time-steps and irregular or random time increments) in a manner that can be exploited by many different application domains. By layering domain-specific C++ classes upon ParaSol, we provide a low-effort and efficient model for domain-specific parallel programming.

Technical Issues in ACES

The central theme of the ACES project is to evolve innovative software systems for massively concurrent computing in heterogeneous collections of computer systems. To this end, we have constructed a layered architecture of four software subsystems that interact closely and enable high performance computing in networked environments. The architecture consists of (a) a communications substrate; (b) a parallelization layer for replicative computations; (c) a threads subsystem for specifying and executing computational tasks; and (d) domain specific library suites for multiple application classes. The challenges in each of these aspects of distributed and parallel computing, and our approaches and solutions to each, are briefly summarized below:

The communications fabric is crucial to any distributed computing system. Scalability, high performance (i.e. high bandwidth and low latency), reliable communications, and systems aspects of implementation are hard issues when deploying parallel applications on general purpose networks. The communications substrate of ACES, viz. Conch, address scalability by introducing the notion of nested topologies, that simultaneously provides for indefinite scaling (subject to system limits) and specification of applications communications patterns. To achieve fast and responsive communications, message passing in Conch is accomplished via special system threads within the application, instead of separate daemons as in PVM. Reliability (message buffers, broken connections) is also handled by these threads, with communication delivery being either provided by or being explicitly implemented over the underlying transport mechanism. To ensure that high communications performance is attained, these threads contain system/OS/network specific code to utilize optimal transmission unit sizes, buffer sizes, and timeout values.

The EcliPSe computation layer in ACES provides two innovative yet pragmatic solutions to important issues concurrent computing on networks, i.e. parallelization and failure resilience. While Conch/CLAM provide for general purpose parallel processing, many applications exhibit structured communication patterns and interactions that can be semi-automatically parallelized in a straightforward manner. EcliPSe provides this capability by enabliing replicative computations to be performed in parallel with minimal user effort in parallelizaaion. Further, given the long running nature of these jobs and the fallibility of networks, EcliPSe, with support from Conch/Clam, automatically restarts failed replications, ensuring consistency of results and synchronization with other parts of the application. Failed processes are restarted at alternative locations and rolled back to a globally consistent state.

In terms of units of computation/scheduling and units of parallelism, the ACES project has adopted a novel and unique approach. Rather than use processes as in PVM, the Ariadne subsystem of ACES enables application programming in terms of lightweight processes. This permits an application to be partitioned in granularity units that are appropriate to the problem at hand, with hardly any loss in efficiency. Further, the potential for overlapped computation and communication is considerably enhanced, by having multiple threads within a single process. The Ariadne system presents a simple programming interface with only seven different library functions, thus avoiding a steep learning curve. Key features of Ariadne include thread-migration across networks of homogeneous machines (currently Sparc nodes), and primitives supporting coordination (in particular, termination) of distributed threads. Moreover, Ariadne has been designed to be substrate-independent, in particular of Conch/Clam, so that it may be used in applications with a variety of communication libraries (e.g., PVM, MPI).

The final layer in the ACES project is ParaSol, the domain-specific library layer for different applications domains. Building upon theEcliPSe idea, ParaSol permits rapid and straightforward assembly of concurrent applications by providing readymade libraries that fit certain classes of application domains. For example, most

particle-in-cell codes possess a standard problem setup structure and a regular cycle of computation and communication, with information exchange occurring along spatial domain boundaries. ParaSol provides template libraries that can automatically handle these phases, but with substitution of parameters for a particular problem size and initial conditions. Underlying Ariadne threads are used for the computation and a combination of EcliPSe and Conch facilities are used for the communication.

Overall the project has been very successful, and our group has graduated four Ph.D. students and one M.S. student between 1996 and 2000, in addition to a number of papers (as submitted in previous reports of progress). Further all four Ph.d. students and the M.S. students were awarded the prestigious Halstead Award in Software Engineering (awarded annually by the Purdue/Oregon Software Engineering Research center for the best work done by a student in the computer sciences program at Purdue University; additionally, the M.S. student was also awarded a Burton D. Morgan award from the Krannert School of Management for his contributions to entrepreneurial work). Over the past year we have focused our efforts on consolidating the software, but the project terminated in March 2001, before the students employed on the project could finish their work for the semester.

Publications

1. *Emulating Parallel Programming Environments in the Harness Metacomputing System*, (preprint) M. Migliardi and V. Sunderam.
2. *Automatic Reincarnation of deceased plug-ins in the Harness Metacomputing System*, (preprint) M. Migliardi and V. Sunderam.
3. *A Repository System with secure file access for collaborative environments*, (preprint) P. Gray, S. Chandramohan and V. Sunderam.
4. *4. On Tailoring Thread Schedules in Protocol Design: Experimental Results,* (preprint), J. Gomez, V. Rego and V. Sunderam.
5. *Multithreading Techniques and Applications,* (in preparation), J. Sang and V. Rego.